

GoldFile

GoldFile

Introduction

GoldFile is a database program. A database program allows you to create and manage a database, which is a collection of information on a related topic. Examples of a database might be a mailing list (a collection of names and addresses), an inventory list (a collection of item names and other information), an index of magazine articles (a collection of article and publication names, dates and topics), or any list of a number of related things. The information about each thing in one of these lists - a person in the mailing list, an item in the inventory, or an article in the index - is called a *record*. The individual bits of information in each record, like a person's name or address in the mailing list, is called a *field*. The collection of all of the records is the database, and the database program lets you add new records, modify existing records, search for records, print reports, and a variety of other interesting and useful functions that you'll learn about in this manual.

GoldFile is a "flat-file" database program that organizes records as rows of information, with the different fields spread horizontally as columns. A simple example of data organized in this fashion is an address book, where each entry appears in the format:

First Name Last Name Telephone Address City State Zip code

All the entries in this database (each is a record) would be displayed on the screen, one after the other. The flat-file format

makes it easy to move through the records and fields of your database ("Name", "Address" and "Telephone" are the fields in this example), since you can scroll vertically to reach any record, and horizontally to reach fields that may extend beyond the boundaries of the screen. For more convenient access to entire records, a full-screen mode can be used to edit a single record at a time.

Fields in a record don't have to be text as in the above example. A field can contain a number, a currency value, a "boolean" value (true or false), a date, or one of several other special types.

Records in the database are displayed on the screen, but they are stored on disk at all times. You can add new records, delete records, and modify existing records. Each time you make a change to the database, the new information is recorded in the database file stored on disk. Storage of the database on disk instead of in memory allows you to use very large databases, and ensures that you don't lose the entire database due to a power failure or other system malfunction.

The records can be ordered by any index that you define, and any number of indexes may be created. In the above example, you might have an index to sort by first name, one to sort by last name, another to sort by zip code, and yet another to go in order of area code (from the Telephone number). As you modify or add to the database, all of the indexes will be automatically updated so that the data always remains in the correct order. Indexes are stored in memory, and must be saved to disk.

There are many other powerful features of *GoldFile* that you will learn about in this manual. Most of the material is covered in a "hands-on" manner, so you will find it easiest to learn if you have the program running and work with it as you read the manual. You will be creating a database while learning how to use the program, so you should have a blank data disk available (or free space on a hard drive) to store the database on.

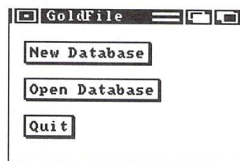
Creating a Database

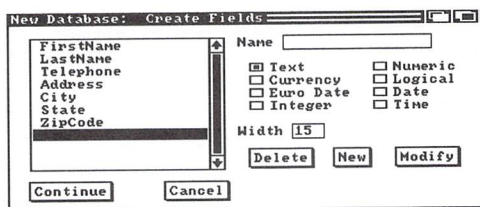
A database consists of two separate parts: the "template" that defines all of the fields in each record, and the data itself. When you first create a database, you must define the template. The template defines the name, type and size of each field in the records, and it needs to be set up only once for a specific database. The data itself consists of the names, addresses, or whatever information is to be stored in the database, and the data is usually updated from time to time. While the data may change, the database template remains the same. The database and the template are stored in separate files on disk: you assign the name for the database file, and the template file has the same name with ".tmp" at the end.

Defining the database template

When you first run *GoldFile* (see the "Getting Started" section of the manual) the first thing the program does is put up a requester that presents three choices: "New Database", "Open Database", or "Quit". Click on "New Database" to tell *GoldFile* that you are starting with a "blank sheet" and not loading in a database that has already been created.

The main window will appear, and, after a few seconds, the *Create Fields* requester will appear. This is where you define the fields making up the database template.





On the left of the requester is an empty list; this is where the field names will appear as you define them. On the right of the requester are various gadgets that let you specify the name, type and size of a field, as well as three buttons at the

bottom that let you delete, create, or modify fields.

Each field in the database must have a name. The field names serve two purposes: 1) they are displayed above the records in the main display, helping you to identify the data, and 2) they are used when specifying expressions for finding or filtering records in the database.

As an example, try defining the simple address book database example. In this database, each record holds information about a person on your mailing list. A business mailing list would probably involve some extra fields, but let's use the example of a personal mailing list where the only information you need is the person's name, address, and telephone number.

It will be useful to index such a database by either first or last name, so we'll use separate fields for these. We could call the fields "FirstName" and "LastName". Field names may be up to fifteen characters long, and should not contain spaces or other punctuation symbols except the underscore (_) character. The other fields in our database will be "Telephone", "Address", "City", "State", and "ZipCode". (This is U.S. specific; in Canada, for example, you might choose the names "Province" and "PostalCode" instead of "State" and "ZipCode".) In this example, all of the fields used will be of the "text" type. Other field types will be covered later.

To create a text field:

- Type the field name - it will appear in the text box labelled "Name" - then press RETURN. The cursor moves to the text box labelled "Width".
- The default width is 9 characters. Change this to the maximum size you think you'll need for the data in this field. For example, the "FirstName" field would probably be adequately served by 15 characters. Delete the "9", enter 15 in the "Width" box, and press RETURN.

- To enter a new field without pressing RETURN in the field width box (to accept the default width, for example), click on the *New* gadget below the field width.
- The field has now been added to the list of fields. Repeat the process for each field in the database.

Typical field names and sizes might be:

Field name	Size
FirstName	15
LastName	15
Telephone	12
Address	40
City	15
State	2
ZipCode	10

- To delete a field, click on the field name in the list and click the "Delete" button.
- To change the size of a field already created, click on the field name in the list, click in the "Width" box, change the number there, and press RETURN.
- Once all fields have been set up as desired, click *Continue* on the requester. The requester goes away, and the file requester appears.
- Select a device (the disk drive holding your blank data disk) and file name to save your database to. The file name "AddressBook.db" might be a good choice for this database. Click *OK* on the file requester. (See Appendix H for details about operating the file requester.) You should have enough room on the storage device to save the entire database; at this point, however, only the template is saved.

Now the database template is defined and saved, and you're ready to add the data itself.

Records

While the template defines the structure of a database, it would be useless without the records, which contain the actual data.

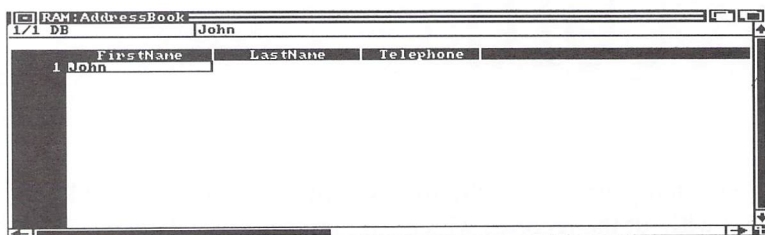
Depending on the database, a record might store information about a person, a company, a transaction, a magazine subscription, a piece of inventory, or anything that you might need to look up.

As you add new records, delete records, or modify the fields in records, the database file on disk is constantly updated to keep track of the current state of the database. If the power goes off or your computer experiences hardware or software failure, you will not lose any records in the database providing the database file on disk has not been damaged. You may lose index information (indexes determine the order the records are sorted in), but you won't have to re-type the data in the records.

The display

If you've just created a database as explained in the previous section, the basic "browse mode" display will appear, but it will be empty - there are no records in the database yet. Before entering some records, you should take note of the basic display. You are currently in "browse mode", which is the default display mode. In browse mode, records are listed one after another in rows down the display, and the fields are laid out across the display in columns from left to right. The fields are labelled, using the names you

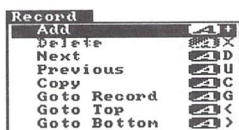
entered when creating the database template, in a row near the top of the display. All of this takes place in a standard window as used by most Amiga programs; it can be depth-arranged, resized, dragged and closed in the usual manner. Associated with the window are standard Amiga-style *Menus* that you select by holding the right mouse button, moving over the menu bar at the top of the window, and releasing the mouse button over the desired menu item. If you are not familiar with these window operations, consult the manual that came with your Amiga. The window also features scroll bars and scroll arrows along the bottom and right edges, which are similar to the ones on Workbench windows: these gadgets let you display different areas of the database, and also indicate which part you are currently looking at.



Adding new records

You are about to add some records to your empty database. You may wish to use entries from your personal address book as examples and as a start towards computerizing your paper-bound information. If not, you can just type in any text at all into the records when following along with the explanations below; the principle is the same whether you're using real data or not. You can always delete these records afterwards, or delete the entire database from your disk, if you wish.

Adding records is always done in a special "add mode", which means that you always follow these steps when adding one or more new records:



1. Enter *add* mode: select *Add* from the *Record* menu.
 2. Enter the new records
 3. Exit *add* mode: select *Add* from the *Record* menu again.
- This is a vital step, as it causes the new records to be merged with the rest of the database.

When you are in *add* mode, the menu item has a checkmark beside it, and several other menu items are no longer available. When you select *Record/Add* again to exit the mode, the new records that you entered are merged with the rest of the database, and all of the menu items function normally again.

Assuming that you have created the template for the address-book database as explained in the previous section, add the first record as follows:

- Select *Record/Add* to enter *add* mode. The message "Add record mode" is displayed in a message box directly above the field name headings.
- Begin typing in the contents of the first field ("FirstName"); you will see the text appear in the text entry box above the field names. Complete your entry by pressing the RETURN key on the keyboard.
- The text you entered for the first field appears in the main display under the "FirstName" field heading. You are now ready to enter the text for the next field, "LastName". Continue to fill in all the fields for the first record in the same way. When you press RETURN after entering the last field in the record, a new record is created for you to fill in in the same way.

You will notice that when entering the text for a field, a text cursor appears in the field entry box above the field names. This is how you fill in new fields, and also how you make changes to existing fields. You can move the text cursor in this box with the cursor keys, and use the DEL and backspace keys to make corrections as you type. (This is *not* a standard Amiga string gadget.)

There are a few more features of the display you should notice at this point. To the left of the field entry box is a pair of numbers that indicate the current record number, followed by the total number of records in the database. Directly below this is a strip that extends the width of the window where messages are displayed; you already saw the "Add record mode" message here. Below this, going down the left side of the display, is a solid area that shows the record numbers of the records being displayed; so far, only numbers 1 and 2 are shown, because only two records exist in the database.

- Enter the rest of the records in the database in the same way as the first record. If you make mistakes, they can be corrected later.

- After entering a few records (about ten or fifteen records will make a reasonably sized database to experiment with), exit *add* mode by selecting *Record/Add* again. The newly entered records are now added to the database.

After you exit *add* mode, you may notice that the records are listed in a different order than they were entered in. This is because the database is always sorted according to the current index. Since you have not created an index of any kind, the default index is used, which sorts records according to the first field. In this case, your address book will be sorted in alphabetical order, by first name. The "indexes" section of the manual shows you how to create your own indexes so that you can sort by any field or combination of fields.

Browsing and editing

Now that you have a database with a few records in it, you can explore browse mode in more detail. Notice the box surrounding one of the fields on the display: this is the *field cursor*, and it indicates which field is the *current field*. You can move the field cursor around using the cursor keys on the keyboard. If you move the field cursor to the right while it is at the right edge of the window, the display scrolls horizontally so that you can edit fields that were not previously displayed. You can redisplay the first fields again by moving the cursor back to the left by the same number of fields; the field names always correspond to the data shown in the window. Moving up and down moves from record to record, and when there are more records in the database than can be displayed in the window, scrolling occurs vertically. You can also move the field cursor directly to any visible field by simply pointing to the field and clicking with the left mouse button.

Editing fields

The contents of the current field always appears in the field edit box, where you can edit it simply by clicking in the field edit box, changing the text, and pressing RETURN. (Pressing the F1 key on the keyboard has the same effect as clicking in the field edit box, and lets you modify records without using the mouse.) When editing a field in the field edit box, you can abort the edit process and leave the field unchanged by pressing the ESC key.

Moving the field cursor and editing fields in the field edit box gives you a casual way to browse through the records of your database, viewing the contents and changing the fields at will. You

can experiment with the editing process by fixing up any mistakes you may have made when entering the records.

Since the records are always sorted by the current index, editing a field can cause the records to re-order themselves. For example, the default index currently used sorts the records alphabetically by the "FirstName" field. Try changing one of the first names to see what happens:

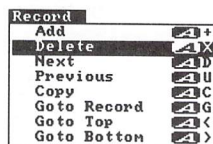
- Move the field cursor to the "FirstName" field of the first record, either by moving it with the cursor keys, or by clicking on the field with the mouse.
- Click in the field edit box, or press the F1 key to edit the field.
- Move the cursor to the beginning of the text in the edit box, and type "ZZ" to change the name.
- Press RETURN to enter the new text into the field; the records will redisplay, and the record you just modified will appear at the bottom of the list.

Selecting ranges

Some operations, like deleting records, can be performed on more than one record in the database simultaneously. You can select a range of records by clicking on the first record in a range, holding the left mouse button down, and dragging the mouse to enclose the desired range of records. The field cursor will expand to enclose all the selected records. For enclosing very large ranges of records, it is often more convenient to use the "shift-click" method of range selection: click on the record at the start of the range, then click on the end of the range while holding down the SHIFT key on the keyboard.

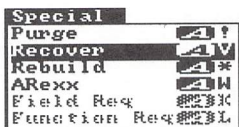
Deleting records

To delete a record or a range of records, select *Delete* from the *Record* menu. If the normal field cursor is showing, just the record that the field cursor is on will be deleted. If a range of records is selected, all records enclosed in the range will be deleted.



Recovering deleted records

When a record is deleted, it will no longer be available, but the data is not physically removed from the database file. Because of this, deleted records may be recovered - a feature that can be most welcome when records are deleted accidentally!



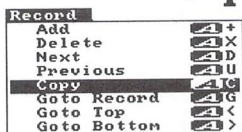
To recover deleted records:

- Select *Recover* from the *Special* menu.
- You will be prompted to enter the date to recover from. Enter a calendar date, as in "Mar-28-90". All records that have been deleted from that date onwards will be restored and will reappear in the database.

Purging deleted records

Since deleted records are still in the database, they are still taking up the same amount of disk space as the active records. If you wish to reclaim the space used by all of the deleted records, you can remove the records completely by selecting *Purge* from the *Special* menu. This operation is final - deleted records cannot be recovered after a purge. You are warned by a requester after selecting the purge command, and given a chance to back out before performing the operation. The purge can take a while with a large database, since all of the records have to be read and re-written to fill in the "holes" left by the deleted records. If you are maintaining a database where new records are added and old ones are deleted regularly, you should perform a purge once in a while to keep the file from becoming unnecessarily large.

Copying records

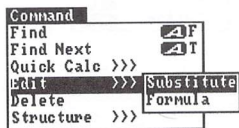


You can make duplicate copies of a single record or a marked range of records. When you select *Record/Copy*, the current record will be duplicated, or, if a range of records is marked, all records in the range will be duplicated.

Mass editing

You have seen that it is easy to change a field in any single record, but there is also a simple way to change every field in an entire range of records, or in all records in the database.

To change the contents of a field in every record in the database:



- Move the field cursor over the field you wish to change. It does not matter for this purpose which record it is in.
- Select *Edit/Substitute* from the *Command* menu; you will be prompted to enter the new value.
- Enter the new contents of the field and press RETURN.

The selected field in every record will be replaced by the new value. If you wish to cancel the operation, press ESC instead of RETURN.

To change the value of a field in a selected range of records:

- Select the range of records by clicking and dragging as described earlier.
- Select *Command/Edit/Substitute*; a requester appears asking you to choose a field.
- Select the field you wish to change by clicking on the field name in the list, then click *Continue* on the requester. If you wish to abort the process, click on *Cancel* instead.
- A prompt appears asking you to enter the new value. Enter the new contents for the chosen field and press RETURN (or press ESC to cancel). The selected field in every record in the range will be replaced by the new value.

NOTE: The mass edit is a dangerous operation in that there is no way to get back the original contents of the affected fields. Make sure you have selected the right field, and that you won't be losing any important data, before performing this command.

Saving and loading

Although the records are saved to disk as you work, you should still save the database to disk often so that the index is kept up to date. To save the database, just select *Save* from the *Project* menu; this will save the database template and the current index information in the existing template file.



If you exit the program without saving after modifying the database, or have a system failure before you get a chance to save, the next time you load the database the records may be mixed up.

Mixed up records can cause deleted records to appear, and good records to be unavailable. If this happens, you can rebuild the indexes and restore the database to its proper condition again by selecting the *Rebuild* item in the *Special* menu.



Using existing databases

When you run *GoldFile* and wish to use a database that you've already created, you should select "Open Database" instead of "New Database" in the initial requester. The file requester will then appear, allowing you to select the database file you've already created. Do not select the ".tmp" template file that has been created by *GoldFile*, but choose the file with the name you selected when you first created the database. After a few seconds, the database

will load and the records will appear in the display.

You can also create or load a new database without disturbing the one you're already working on; *GoldFile* lets you work with more than one database at a time. Each database will be in a separate window, and you can switch between them as you would with any separate windows in the system: just click on the window for the database you wish to use. To create a new database in another window, select *Project/New*. The *Create Fields* requester will appear to let you define the new database. To open a new window and use a database that has already been created, select *Project/Open...* The *file* requester will appear, letting you choose the database file you wish to open. In either case, a new window, slightly shorter and positioned just below the top of the first one, will appear with the new database. You can open as many database windows as the amount of free memory in your computer will allow. You can close a database window using the window close gadget, or by selecting *Project/Close* when the window is active. Make sure you save the database before closing the window.

Other field types

The sample address book database you've created uses only text fields. For other kinds of database, other field types are sometimes required. Field types are selected when you create a database template, by clicking on the appropriate *Field Type* gadget before clicking on the *New* gadget to add the field. A discussion of the various field types follows:

Text

Text fields can hold any characters you can type from the keyboard: letters, numerals, punctuation, and special symbols. The maximum number of characters stored in a text field is limited by the field width selected when the database template was created. The greater the width of a text field, the more disk space will be used by the database, since more characters must be stored for every record. Other field types use the same amount of memory regardless of the field width, which is only used for display purposes.

Numeric

Numeric fields hold positive or negative numeric values, which can be integers or floating point real numbers with a decimal point. The field size that you specify must be large enough to contain at least the integer part of the number; values too large to fit in the

allotted field width will be displayed as a series of asterisks. Digits after the decimal point are cut off (truncated) to fit within the field's width, but the actual value (accurate to about six decimal places) is displayed in the text edit box when you move the field cursor over the field - this is the value used in calculations. Unlike text fields, the field width specified for a numeric field does not affect the size of the database file; it affects the amount of room used in the display only.

Numeric fields can be used in calculations using the *Command/Edit/Substitute* or the *Command/Quick Calc* facilities, which are explained in the "Advanced Features" section of the manual.

Currency

The currency field is a numeric field that displays its value in dollars and cents. The value is always shown truncated to two decimal places, but it is internally stored with greater accuracy. The field width that you set for a currency field should be large enough to hold the required number of dollar digits, the dollar sign, the decimal point, and the two cents digits. For example, a currency field large enough to properly display amounts in the thousands of dollars (up to \$9999.99) would have to be eight characters wide. Negative values require an additional character of width to display the minus sign (-). When in doubt, leave a few extra digits in the width - the amount of disk space is not affected by the field size chosen for numeric fields, as it is for text fields. Currency fields can be used in calculations in the same way as numeric fields.

Logical

Logical fields are numeric fields that display as "FALSE" if the value is zero, or "TRUE" otherwise. The field width must be at least five characters, in order to show the text "FALSE". Like numeric fields, the size that you specify for the field width does not affect the size of the database file. When entering values into logical fields, you can type "F" or "0" for false, or "T" or "1" for true - anything you type into these fields is translated to a value of zero or one. Although they display as either true or false, when used in calculations logical fields retain a complete numeric value, and can be used in the same way as regular numeric fields.

Date

The Date field stores a calendar date and displays it in the format "Mmm-DD-YY", for example "Mar-12-90". The default field

width of nine characters is the minimum size required to display a date. Date fields are actually stored in the same way as numeric fields, with the stored value representing the number of days since Jan-01-78. You can enter dates into these fields in the displayed format, with or without the dashes between the month, day and year. A special feature of the date field allows you to enter today's date by simply typing an asterisk (*) into the field.

Euro-Date

The Euro-Date field works in the same way as the regular date field, but displays the date in the standard European format, which is "DD.MM.YY", as in "12.03.90".

Time

Time fields are similar to date fields, but display a time of day. Times are entered as "MM:SS", as in "4:30". Times can be entered in either military-style 24-hour time (e.g. "16:47"), or by specifying "PM", as in "2:47 PM". As with the date field, an asterisk entered into a time field translates as the current time. The minimum field width required to display a time field is seven characters.

Integer

Integer fields are similar to numeric fields, but can only hold integer values: any fractional component of a number is truncated. Integer fields are useful for things such as part numbers, current number of products in stock, etc., where fractional numbers are not allowable. You can perform "Quick calc" calculations on integer fields (see "Advanced Features"), but all results will be rounded to the nearest integer. The range of allowable integers is from about negative two thousand million to positive two thousand million (a computer 32-bit "long-word").

Full-Screen mode

You have so far been viewing and editing your database in "browse mode", which lets you see a number of records on the screen at the same time. In *full-screen* mode, only one record can be displayed at a time, but the information is presented in a more orderly fashion. In browse mode, all of the fields in a record may not be visible at once because the fields may take up too many columns to fit in the display. In full-screen mode, the fields in a record are laid out over several rows, so the entire record can be seen at once.

Another advantage of full-screen mode is that you can define your own *screen formats*, defining which fields from the record are displayed on the screen, where they appear, and even how they are labelled. The text used in a screen format need not correspond to the field names, or there doesn't need to be any extra text at all.

For example, you might wish to edit records in address-label format, without any field names at all.

The default screen format

You don't need to define your own screen format to use full-screen mode; a default screen format displays the record with fields labelled by their field names, with one field per row.



To try out full-screen mode with the default format, turn off browse mode by selecting the *Browse* item in the *Project* menu. The screen

DF0:Address	
1/4 DB	Jane
FirstName:	Jane
LastName:	Doe
Telephone:	555-4321
Address:	
City:	
State:	
ZipCode:	

will change to show only the current record. On the left of the display are the field names, one per row, and to the right are the contents of the corresponding fields in the record.

You can edit fields in the same way as in browse mode: the field cursor can be moved to indicate different fields, and you can click in the field edit text box or press the F1 key to edit the contents of the current field. If you cursor down off the last (bottom) field in the record, the next record will be displayed. If you cursor up off the first (top) field, the previous record will be brought in. You can move directly to the next or previous record by using the *Next* and *Previous* items in the *Record* menu.

Adding new records can also be done in full-screen mode in the same way as browse mode: select *Record/Add* to enter *Add* mode, and enter the data into the fields of the record, pressing RETURN after each field. After the last field in the record has been filled, a blank screen will appear for you to enter the next record. After you have finished entering new records, select *Record/Add* again to leave *Add* mode and continue editing or viewing records.

Experiment a bit with the full-screen mode by adding new records, and viewing and editing the records in your database. You can switch between full-screen and browse modes at any time by selecting the *Project/Browse* menu item once again - a checkmark appears beside the menu item when browse mode is selected.

Custom screen formats

There are times when the default screen format is not ideal for your database or your data entry requirements. Sometimes a database contains fields that are not used when entering records initially. As an example, suppose you had a product inventory database in which each record included a product's name, a product code, cost, retail price, number in stock, optimal number to stock, date last ordered, and perhaps some other accounting and stockroom information about the product. When you are first entering your inventory into the database, you wish to start with just the name, code, cost and number in stock for each product. The rest of the fields will be filled in over time as you order new items, see how they are selling, etc. When entering the products, you don't want to be distracted or slowed down by unimportant fields. By defining a screen format that contains only the fields of interest, with meaningful text to label the fields, you only need to enter the required data for a product before moving on automatically to the next record.

As a more familiar example, consider our standard address book database. Screen formats let us present the data in a more natural way. Names and addresses might be commonly entered in the form of an address on an envelope. There are no field names, because it is obvious that the first line contains the name (first name followed by last), the next line the street address, followed by the city, state/province, and the zip/postal code on the last line. Setting up such a screen format is a simple matter.

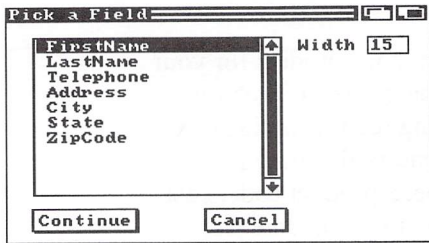
Like filters and indexes, you can define any number of screen formats, and the screen formats are saved with the database. Existing screen formats can be modified or deleted, and you can switch from one screen format to another at any time (or exit full-screen mode altogether and return to browse mode).

Creating a Screen format

Screen format creation is done entirely through the *Screen* menu (next to the *Filter*) menu. Create the simple address format as follows:

- Select *Screen/Create*; a requester appears.
- Enter a name for this screen format, and modify the values for "Cols" and "Rows" to numbers that will fit the new format. For this example, use five rows and 50 columns. Click *Continue* on the requester.

Enter Screen Info	
Name:	Data Entry
Num Cols	50
Num Rows	5
Continue Cancel	



- You are now using the screen format editor. You can add either text or fields anywhere on the screen, within the bounds set by the number of rows and columns you selected in the requester.
- To enter a field: select *Screen/Add Field*; a requester appears. Select the

desired field from the requester and click *Continue*, or just double-click on the field name. For this example, pick the first field, "FirstName".

- Add the next field, "LastName", in the same way. Press RETURN to move the cursor to the second row of the screen format, and enter the next field, "Address". On the next line, add the field "City", followed by a comma, followed by the state/province field. On the final line, add the zip/postal code field.
- Select *Screen/Create* again to complete the screen format and return to the database.

There are two ways to use a screen format that has been created:

- 1) Select *Screen/Use* and pick a screen format from the list that appears, or
- 2) Select *Project/Browse* to enter full-screen mode using the screen format previously used.

Try using the new screen format on your database. You can switch back to the default screen format by selecting *Screen/Use*, and picking "Default" for the screen. To return to browse mode, select *Project/Browse*.

You can delete a screen format using *Screen/Delete*, but, as with indexes, you cannot delete the screen format currently in use. If you get the message "Can't kill current screen", select a different screen format and try the delete operation again.

To modify an existing screen format, select *Screen/Modify*. After selecting the screen format you wish to modify, you will enter the screen editor with the selected screen displayed. You can add new fields, delete existing fields, and add or edit text on the screen. Select *Screen/Create* to return to the database.

Indexes

An index determines the order that the records in the database are listed in. The default index sorts the records in ascending alphabetical order by the first field, but you can define any number of additional indexes that sort the records in other ways. For example, for an address book database, you might want one index to sort by last name, and another to sort by area code (phone number). By selecting one index or another, the records instantly reorder themselves to correspond to the new index.

Indexes can consist of multiple levels of sorting priority. For example, in the index that sorts by last name, you may wish to have all records that have the same last name to be sorted by first name. In other words, "Smith, Charles" should come before "Smith, Wayne". Taking this one level further, you may want all records with the same last name and first name (unlikely, but certainly possible in a large database) to be in order of telephone number area code. Indexes may be created with any number of levels, from a single one up to the number of fields.

All indexes are stored in memory as you use the database, and are constantly updated as you modify the database. The indexes are saved with the database when you select *Project/Save*.

Creating an index

When creating an index, you need to tell *GoldFile* which

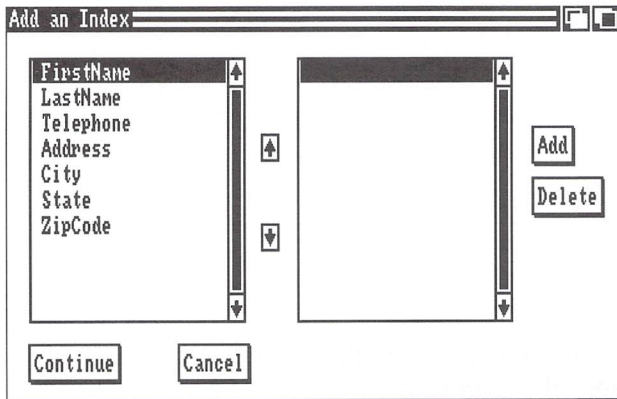


field to sort by, and whether to sort in ascending or descending order. Multi-level indexes are created by specifying several fields. All of this is made simple by the *Edit Index* requester.

Creating an index

To create an index:

- Select the *Index/Create* menu item; you will be prompted to enter a name for the index.



- Enter a meaningful name for the index (e.g. "Last name" and press RETURN); the *Edit Index* requester will appear.
- The *Edit Index* requester displays two lists: the one on the left lists all of the database's fields by name, and the one on the right is initially

empty. To select the field used for the index, click on the field name on the left-hand list to select it, then click on the *Add* button (double-clicking on the field name is a shortcut that has the same effect). The field is removed from the left-hand list and added to the right-hand list.

- The small up-arrow to the right of the field name indicates that this field will be sorted in ascending order (A to Z or smallest to largest). To change this to descending order, click on the down arrow button between the two lists.
- To add another level to the index, just add another field to the right-hand list in the same way. You can add as many levels as there are fields.
- To make corrections, you can remove a field from the right-hand list: select it (if it's not already selected), then click on the *Delete* button. The field will be moved back to the left-hand list.
- When the index is arranged to your satisfaction, click on the *Continue* button at the bottom left of the requester. To abandon the creation of the new index, click *Cancel* instead.

Using an index

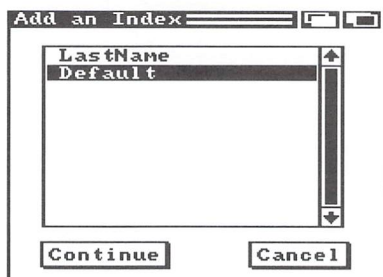
The records in a database will always be sorted in the order of one index or another. The index being used is called the *active* index.

Creating an index does not automatically make the new index active. You have to explicitly select an index to make it active.

To use an index:

- Select the *Index/Use* menu item; the *Use Index* requester will appear.
- Select an index from the list of index names, then click *Continue* on the requester. Double-clicking the index name has the same effect.

The records will now be ordered by the newly selected index.



Deleting an index

If an index is no longer required, you should delete it to free memory and speed up database operations.

To delete an index:

- Select the *Index/Delete* menu item; the *Delete Index* requester appears.
- Select the index to delete from the list and click *Continue* on the requester, or just double-click the index name.



NOTE: You cannot delete the currently active index.

Filters, Searches and Sets

Filters let you look at only those records in a database that match certain criteria. For example, in the address book, you may wish to view only those people living in California, or only those that have a certain area code. Criteria may be more complex, like specifying only those customers in a specified state that have made a purchase of more than 1,000 dollars in the last six months.

Filters work somewhat like indexes in that a number of separate, named filters may be specified, and you can select any one of them. When you "apply" a filter, the database is instantly limited to those records matching the filter criteria - only those records are displayed and available. You can just as easily "forget" the filter and switch back to the entire database. By applying filters one after another, you can pare down a database by being more and more selective about the records included. For example, consider the database of customers mentioned above. Say the database contains the fields "Name", "Address", "State", "Date", and "Purchase".

Applying one filter might display only those customers in California (State field contains "CA"). Applying the next filter could extract the records with a date more recent than December, 1989. Applying still another filter, the remaining records would be only those with a value greater than 1,000 in the "Purchase" field. Searching for a specific record or records in the database is done

with the *Command/Find* and *Command/Find Next* menu items. Searching does not modify the displayed database records the way filters do, but the required "Find formula" works in exactly the same way as the expressions that you enter as filter criteria. The sections below on Filter criteria and wild-cards apply equally to filters and search operations.

Filter criteria

When you create a filter, you must specify the criteria for records to match. This can be entered in the form of a simple expression using a field name, an operator, and a constant value. (More complex expressions involving built-in functions can be used as well; these are covered in the "Advanced Features" section of the manual.)

For example, in the above case where we are checking for "CA" in the "State" field, we would use the expression:

State="CA"

Since we are checking for the state name to be a specific piece of text, we use the "=" operator to mean "equal to". Since "State" is a text field, we must compare it to a text string. Enclosing the letters "CA" in quotes signifies a text string. Checking for values in a numeric field would require no quotes. The State fields are checked, character by character, with the string in quotes, ignoring the case of the letters ("CA" would match with a State field containing "Ca" or "ca").

Besides checking for a specific text string or value, you can also use other operators to make different tests. This is a list of all operators:

- <> Not Equal
- = Equal
- > Greater than
- < Less than
- >= Greater than or equal to
- <= Less than or equal to

In the above example, the expression to check for "Purchase" greater than 1,000 would be:

Purchase>1000

Using date fields

When you wish to compare the values of date fields in a filter expression, you can use the built-in function DATE to convert

month, day and year values to the numeric value as stored in a date field. Continuing with the same example, we wish to check the "Date" field for dates since, say, January 26, 1990. This could be done with the following Filter expression:

Date>DATE(1,26,90)

The values (arguments) given to the date function are the month, day, and year, respectively. Other date functions let you find the month, day or year from a date value, so you could search for dates in the year 1988, for example, with the expression "YEAR(Date)>88".

Creating a Filter

When you create a filter, the new filter is added to the list of available filters, but it is not actually applied to the database until you explicitly give the *Filter/Apply* command. To create a new filter:

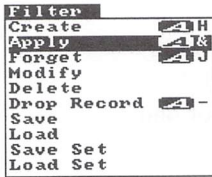
- Select *Filter/Create*, the first item in the *Filter* menu; you will be prompted to enter the name of the filter.
- Type a name for this filter (e.g. 'CA residents') and press RETURN; you will be prompted to enter the filter.
- Type in the filter expression (e.g. 'State="CA"') and press RETURN. If you use a nonexistent field name or enter an invalid expression of some kind, an error message will be displayed and the input will not be accepted; if this happens, edit the expression and try again.
- If you're not sure of the exact field names used in the database, you can choose a field name from a list instead of typing it in. While entering the filter expression, select *Field Req* from the *Special* menu (the last menu). A list of field names will appear. Select a field from this list by double-clicking on it. The selected field name will be inserted into the filter expression.
- To abort the filter create operation, press the ESC key while entering the expression.

Once successfully created, the filter will be available for use. Filters are not automatically saved with the database, but each filter can be saved in a separate file and loaded as needed.

Applying filters

When you apply a filter, the entire database is searched for records that match the filter conditions as specified in the filter's expression, and only those records are displayed. This set of

records (called a "set") is now used in all subsequent database operations. You can continue to edit records, change indexes, and apply new filters to the records as if you were working with an entire database. To apply a filter:



- Select the *Apply* menu item from the *Filter* menu; the *Choose Filter* requester appears listing all of the filters by name.
- Select a filter from the list and click *Continue* on the requester, or just double-click on the filter name. The filter will be applied to the database, and only those

records meeting the filter criteria will be displayed from now on.

- To "forget" all filters and display the entire database, select *Filter/Forget*.

Notice that when you apply a filter, the record number display in the top left text box in the window changes. Instead of "DB" appearing next to the record number information, indicating an entire database, the word "Set" appears. This lets you tell at a glance whether you're working with the entire database, or just a set of records. When you select *Forget*, the display reverts to "DB" and all records are available again.

Wildcard characters

Special characters in filter expressions let you specify some criteria more generally. The asterisk (*), when used within a text string, means "any number of characters". It is called a "wildcard" because it matches with anything. As an example of where this might be useful, suppose you wish to create a filter that extracted all records in your address database that have a certain area code in the telephone number (field name "Telephone"). You only want to take the first three characters of the telephone number into account when making the comparison. This can be easily accomplished using the asterisk wildcard character as follows:

Telephone="415*"

This would filter all records with telephone numbers beginning with "415", regardless of the characters in the rest of the "Telephone" field.

You are not limited to using the asterisk at the end of a string, but it can represent a number of characters at the beginning or the middle of a string as well. For example, if you had a simpler address database where the first and last names in the records were stored

in a single field, you could search for a specific name and ignore any middle initials with an expression like this:

```
Name="Jim * Smith"
```

Another wildcard character is the question mark (?), which means "any single character". Question mark wildcards may be used singly, in groups, or repeated in various places throughout the search string. As an example of using both types of wild-card characters, suppose you wanted to search your simple address database for the name "Cathy McBride", but were not sure if the name was spelt "Kathy MacBride", "Cathy McBryde", or any one of a number of other possibilities. To cover most possibilities, you could specify the search string or filter expression as follows:

```
Name="?athy M*Br?de"
```

Record searching

The same kind of expression used in creating a filter can be used to locate records meeting specified criteria, without changing the database in any way. Searches always take place starting from the record that the field cursor is placed at, and each record is checked in sequence from that point onwards.

To find a record:

- Position the field cursor on the first record in the database.
- Select *Find* from the *Command* menu; you will be prompted to enter the find formula.
- Enter an expression (e.g. 'FirstName="Joe"'), and press RETURN.
- If a record is found that matches the find expression, the message "Found" will show in the message box, and the field cursor will be positioned on the first field of the matching record. If no matching record is found, the message "No records found" will be displayed.
- To perform the same search on the remainder of the records, select *Command/Find Next*. You can repeat this command until no more matches are found.

Saving and loading filters

If you wish to use the same filter the next time you load the database, you must save the filter and load it again when you need it. Each filter is saved as a separate file.

To save a filter:

- Select *Save* from the *Filter* menu; a requester appears listing the available filters by name.
- Choose the filter you wish to save by clicking its name and selecting *Continue* on the requester. As a shortcut, you can just double-click on the filter name. The file requester will appear.
- Choose a file name to store the filter under and click *OK* on the file requester. You can save the filter on any disk or storage device, but to avoid confusion it is wise to store the filter in the same directory as the database it pertains to.

To load a filter:

- Select *Load* from the *Filter* menu; the file requester appears.
- Choose the filter file you wish to load and click *OK* on the requester. The filter will be added to the list of available filters. If a filter of the same name already exists, an error message will appear in the message box and the filter will not be loaded.

Sets

Once you've applied one or more filters, the records that remain can be saved as a *set*. A set is a collection of records in a database, but the file that you save as a set does not actually contain the records themselves: they are in the database file only. The set consists of pointers to those records.

Saved sets can also be used as a merge file for *TransWrite*, so that you can send a form letter to specific people stored in your database.

Creating a set

Besides using filters, you can also manually drop selected records from the set.

To drop a single record from the current set:

- Place the field cursor on any field within the record you wish to remove from the set.
- Select *Filter/Drop Record*; the record disappears from the display.
- If you select *Filter/Forget*, the dropped record, along with all other records in the database will be redisplayed. Dropping or filtering records does not delete them from the database, only from the currently displayed set.

To drop a range of records from the set:

- Mark a range of records by clicking and dragging with the mouse.

- Select *Filter/Drop Record*; all records in the marked range disappear from the display.

Saving and Loading a set

A saved set can be recalled again for further database operations, or can be used as a "merge file" in *TransWrite*. An example of a set might be all invitees to a wedding or party. You would first generate the set by filters or by manually dropping records of people who are not to be invited. You would then save this set. The saved set could be used to send out invitations, or could be loaded again into *GoldFile* for further database work.

To save a set:

- Select *Filter/Save Set*. The file requester will appear.
- Select a file name for the set and click *OK* on the requester.

To load a set:

- Select *Filter/Load Set*. The file requester will appear.
- Select a saved set file from the requester; the records in the saved set will be displayed.

Since a set is just a collection of pointers to actual records in the database, set files are only valid for as long as the basic layout of the database remains unchanged. If you pack the records in a database (removing all deleted records), previously created sets from the database will no longer be valid.

Reports

Printed reports are a concise way of displaying information in a database. **GoldFile** reports consist of a line for each record, with fields lined up in columns, similar to the "browse mode" display of the database.

When you define a report, you select which fields to print, which field goes in which column, and the column width for each field. You can also specify column headings, or use the field names as column headings. Breakpoints and totals let you extract and present information in a more organized way. All of this information is called the *report definition*.

As with filters, indexes, and screen formats, you can create multiple report definitions, each with a name that you assign. Once created, any report definition can be modified, deleted, or printed. When you print a report, a line is generated for each of the records in the current set or database. All report definitions are saved with the database template, so once a definition has been created, you can use it at any time in the future to generate the printed report. Reports can also be sent to a disk file instead of to the printer, and used in a word processor (*TransWrite*) or desktop publishing program (*PageSetter II*) to be included in more elaborate documents.

Creating a Report Definition

To create a report:

- Select *Report/Create*. You will be prompted to enter the name for the report. Enter a name and press RETURN. To abort the process, press the ESC key instead.
- The *Create Report* requester will appear. On the left side is a list containing the names of all fields in the database. When you double-click on a field name, that field appears in the right-hand list.

Add a Report

Left List:

- FirstName
- LastName
- Telephone
- Address
- City
- State
- ZipCode

Right List:

Width: 000

First Col: 1

Line Break ☐

Form Break ☐

Total ☐

Add Delete

Continue Cancel

Col Title

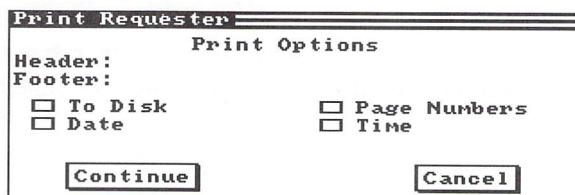
- The order of the fields appearing in the right-hand list determines their position along each line in the report. When you add a new field, the "Last Col" display shows how far along the page that field will reach, in characters.
- The field width is used to determine the number of characters used by each field, but you can change this value by modifying the "Width" text box.
- The field name is used to determine the heading to put above each field's column, but you can change the heading by editing the text in the "Col title" text box near the bottom of the requester.
- To delete a field from the right-hand list, click on the *Delete* gadget.
- When the report definition is set up to your satisfaction, click the *Continue* gadget on the bottom left of the requester. To abort the create operation, click on *Cancel* instead.

Printing a Report

To print a report:

- Select *Report/Print*; the *Print* requester appears.
- The print requester presents a number of options for printing

your report. After setting up the options to your requirements (see below), click *Continue* on the requester. To abort the operation, click *Cancel* instead.



These are the settings you can make on the Print requester:

Header: This is the text printed at the top of each page of the report.

Footer: This text is printed at the bottom of each page of the report.

To Disk: Selecting this gadget brings up the file requester, which lets you choose a file to print the report to. The resulting ASCII file can be imported into a word processor or desktop publishing program and incorporated into a larger document.

Date: When selected, the date will be printed on each page.

Time: Prints the time on each page.

Page Numbers: Prints page numbers on each page.

Breakpoints and Totals

When creating a report definition, there are special gadgets that let you select options for each field. There are two kinds of breakpoints, and a *Total* gadget.

Breakpoints let you split records up into groups on separate pages. When a field is selected as a breakpoint field, a form-feed (page eject) is generated every time the contents of that field changes from one record to the next. As an example of using breakpoints, suppose you wanted to print out your address book database by province or state, with each province or state on a separate page (or pages). You would first choose an index that sorts by state (a second-level sort could be done on the last name). By setting the *Form Break* gadget for the "State" field, the first record of each new state would be printed at the top of a new page. The *Line Break* gadget works in the same way, but it just skips a single blank line instead of ejecting the whole page at every breakpoint.

The *Total* gadget only applies to numeric fields, and causes a total to be calculated and printed whenever a breakpoint is encountered. The total is the sum of the field's value in all records since the last

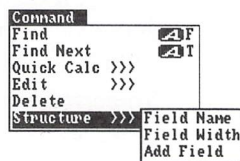
breakpoint. Totals are printed below the related fields, and will be the last printed line on the page (other than the footer at the bottom). Totals are only generated for page breaks, not line breaks.

Advanced Features

Modifying a database template

The database template defines the structure of the database as it is stored on disk. Unlike the data in the database which you add to and modify all the time, the template definition usually stays the same for a given database.

There are times, however, when you may find that you need a field to be larger, or you discover that you need an additional field that you didn't think of when you created the database. Or you may just want to change the name of a field. You can make such changes to a database template after it has been created through the use of the *Structure* menu item in the *Commands* menu. This menu item has several submenus, which let you change the name of a field, change the width of a field, or add a new field.



Since the structure of the database can be changed with these commands, *GoldFile* may have to create a new database using the newly modified template, copy all records from the existing database to the new one, and finally delete the old database file. This procedure will take place if you change the width of a text field or if you add a new field, and it may take some time, especially for large databases. Also, you must have enough room on your disk to store the old and new copies of the database.



To change the name of a field:

- Move the field cursor over the field whose name you wish to modify, in any record.
- Select *Command/Structure/Field Name*. You will be prompted to enter the new name. (If a range of records was selected, you will first be prompted to select which field to change from a list of field names.)
- Edit the field name in the text edit box to the name you wish to use, and press RETURN. To abort the name-change operation, press the ESC key instead.
- The field name will be changed instantly.

Changing the width of a field is done in the same way, except that you enter a new width instead of a new name. If you change the width of a text field, a new database will be created, and all records will be copied to it. If there is not enough room on the disk to store both copies of the database during this process, an error message will appear and the operation will fail.

To add a new field, select *Command/Structure/Add Field*. A requester similar to the one used to create a database template will appear. Define the new field by entering its name, selecting its type, and entering its size in the usual way. Click *Continue* on the requester to begin the add operation; a new database will be created, and all records will be copied to it. The newly created field will be blank (or zero) in all records.

Quick Calc

The *Quick Calc* menu item in the *Command* menu has a number of submenus to perform various statistical calculations on a numeric field in a range of records.

To perform one of the calculations for all records in the current set or database:

- Move the field cursor over the numeric field you wish to calculate, in any record.
- Select the desired calculation from the *Command/Quick Calc* menu by choosing the appropriate submenu.
- The result will be displayed in the display area.

To perform a calculation on a numeric field in a selected range of records:

- Mark the range of records by clicking and dragging with the mouse, or clicking at the start of the range and shift-clicking at

the end.

- Select the desired calculation from the *Command/Quick Calc* menu by choosing the appropriate submenu.
- A requester appears to let you select a field. Choose the numeric field you wish to perform the calculation on by double-clicking the field name.
- The result will be displayed in the display area.

(All calculations work with numeric fields only, except for *Count*, which merely indicates the number of records involved in the calculation.)

Formula substitution

The *Edit/Formula* submenu item in the *Command* menu is a powerful feature that lets you change the fields in a number of records based on a calculated mathematical expression involving the fields in each record. This is similar to the calculated cells in a spreadsheet, where the value shown in the cell can be calculated from a formula based on the contents of other cells. An example of such an expression might be as simple as increasing the value in a field by a certain percentage in all records, or it might be a more complicated expression involving mathematical functions, and using other numeric fields as variables.

As a simple example of using a substitution expression, suppose you wanted to increase the value of a field called "Salary" in all records by ten percent. You could do this as follows:

- Place the field cursor over the "Salary" field in one of the records, or select a range of records in the database.
- Select *Command/Edit/Formula*. If a range of records was selected, a requester will appear to let you select a field to perform the calculation on; pick a field from the list by double-clicking its name. If no range was selected, the calculation will be performed on the field that the field cursor was on, for all records in the set or database.
- You will be prompted to enter the formula. Type "Salary*1.1" and press RETURN.
- The Salary field in all records, or in the selected records, will increase by ten percent (the original value, multiplied by 1.1).

You can use the values from other fields in the same record in the calculation as well. You can type the field names in directly, or use the *Field Req* item in the *Special* menu to help you choose a field

name from a list.

For example, suppose there are fields in your database called "Width", "Height" and "Depth", and another one called "Volume" which you wish to calculate from the other three. You would simply move the field cursor over the "Volume" field in one of the records, select *Command/Edit/Formula*, and enter the formula: "Width*Height*Depth". While entering a formula, you can select *Field Req* from the *Special* menu and double-click on a field name in the list instead of entering the name manually, in the same way as in Filter and Search expressions.

Built-in Functions

Expressions used for Filters, Searches, and Substitutions can be more complex than simple mathematical operations or field comparisons. You can build complex expressions consisting of a number of operations, using parentheses to direct the order of operations in standard algebraic notation.

You can also make use of a number of built-in functions, which perform a variety of mathematical, statistical, logical, and time/date operations. An example of an expression involving fields simply named "F1", "F2", and "F3" might be:

$$(\text{SQRT}(\text{F1})+100)*\text{SIN}(\text{DEGTORAD}(\text{F2}))/\text{AVG}(\text{F1},\text{F2},\text{F3})$$

While it is doubtful that the above expression could serve any useful purpose, it shows the use of built-in functions in expressions. Its effect would be to take the square root of the value in field "F1" using the SQRT function, add 100, multiply this by the Sine of field "F2" (converted from degrees to radians using the DEGTORAD function), and divide the result by the average of the three fields - the average being calculated by the convenient AVG function.

When you wish to put a function in an expression, you can look up the function you need to use from the list below and type it in, or you can select a function automatically from a list using the *Special/Function Req* menu item. You can look through the list of available functions (they appear in alphabetical order) and select the one you want by double-clicking its name in the list. The function will be inserted into the expression, ready for you to type in the arguments to the function, if any.

Expressions can be numeric, boolean, or just simple values. These are defined below.

Numeric Expression:

- any formula which reduces to a numeric value
- numeric expressions may make use of the four basic mathematical operators (addition (+), subtraction (-), multiplication (*), and division (/)) as well as the built-in math functions.
- numeric expressions may be used where boolean expressions are expected. If the expression reduces to a value of zero (0), it is considered FALSE. Otherwise, it is considered TRUE.

Boolean Expression:

A simple boolean expression makes a comparison between two values, either calculated or constant. The result of the comparison is either TRUE or FALSE. Boolean expressions make use of logical operators to perform their comparisons. Examples of the available logical operators are:

$a = b$	TRUE if a is equal to b
$a > b$	TRUE if a is greater than b
$a < b$	TRUE if a is less than b
$a \geq b$	TRUE if a is greater than or equal to b
$a \neq b$	TRUE if a is not equal to b

Complex boolean expressions can be built up from simple expressions through the use of the boolean functions (described later in this section). For example:

`AND( F1 > 5, F1 < 10 )`

This expression is TRUE if the value in field "F1" is between 5 and 10.

Boolean expressions may also be used where a numeric expression is expected. When used in this manner, a boolean expression has a value of one (1) when it is TRUE and a value of zero (0) when FALSE.

Value:

- a numeric constant, e.g. 5.6
- a field name referring to a numeric field containing a numeric value
- any expression (numeric or boolean) which reduces to a numeric value

List:

- a series of values or field names separated by commas

e.g. (1,7.2,F1,Amount)

Statistical Functions

AVG(List)

Returns the average of all numeric values in *List*.

COUNT(List)

Returns the number of items in *List* which have a numeric value.

MAX(List)

Returns the largest value in *List*.

MIN(List)

Returns the smallest value in *List*.

STDEV(List)

Returns the standard deviation of all numeric values in *List*.

VAR(List)

Returns the variance of all numeric values in *List*.

Boolean Functions

Boolean functions allow you to perform "decision making" in your database calculations. These functions treat values as either TRUE (non-zero) or FALSE (zero). In conjunction with the IF function, these functions allow you to present different results based on the status of various fields in the record.

AND(List)

Returns TRUE if all items in *List* are TRUE. Otherwise, returns FALSE.

OR(List)

Returns TRUE if any item in *List* is TRUE. Otherwise, returns FALSE.

NOT(Value)

Returns TRUE if *Value* is FALSE. Returns FALSE if *Value* is TRUE.

XOR(List)

TRUE()

Returns 1.

FALSE()

Returns 0.

IF(Boolean Expression, True Expression, False Expression)

If the *Boolean Expression* reduces to TRUE, returns the value of *True Expression*. Otherwise, returns the value of *False Expression*.

Returns TRUE if there is a mixture of TRUE and FALSE items in *List*. If items in *List* are either all TRUE or all FALSE, returns FALSE.

General Math Functions

ABS(Value)

Returns the absolute value of *Value*.

E()

Returns the value of the constant 'e' (2.7182818285).

EXP(Value)

Returns e raised to the power *Value*.

FAC(Value)

Returns the factorial of *Value*.

FIB(Value)

Returns the fibonacci number of *Value*.

INTEGER(Value)

Truncates the fractional part of *Value* and returns only the integer portion.

LN(Value)

Returns the natural log (log base e) of *Value*.

LOG(Value)

Returns the log base 10 of *Value*.

LOGA(Base, Value)

Returns the log base *Base* of *Value*.

MOD(Value, Divisor)

Returns the remainder of *Value* divided by *Divisor*.

POWER(Value, Power)

Returns *Value* raised to the *Power* power.

RAND()

Returns a pseudo-random number between 0 and 1.

SIGN(Value)

Returns 1 if *Value* > 0, -1 if *Value* < 0 or 0 if *Value* = 0.

SQRT(Value)

Returns the square root of *Value*.

SUM(List)

Returns the total of all values in *List*.

Trigonometric Functions

ACOS(Value)

Returns the arc cosine of *Value*.

ACOSH(Value)

Returns the arc hyperbolic cosine of *Value*.

ASIN(Value)

Returns the arc sine of *Value*.

ASINH(Value)

Returns the arc hyperbolic sine of *Value*.

ATAN(Value)

Returns the arc tangent of *Value*.

ATANH(Value)

Returns the arc hyperbolic tangent of *Value*.

COS(Value)

Returns the cosine of *Value*.

COSH(Value)

Returns the hyperbolic cosine of *Value*.

DEGTORAD(Value)

Converts *Value* from degrees to radians.

PI()

Returns the value of the constant 'pi' (3.1415926536)

RADTODEG(Value)

Converts *Value* from radians to degrees.

SIN(Value)

Returns the sine of *Value*.

SINH(Value)

Returns the hyperbolic sine of *Value*.

TAN(Value)

Returns the tangent of *Value*.

TANH(Value)

Returns the hyperbolic tangent of *Value*.

Time Functions

DATE(Month, Day, Year)

Returns the number of days between January 1, 1978 and the date indicated by month number *Month* (1 to 12), day number *Day* (1 to 31) and year number *Year*. This is the value stored in a date field to

represent a date.

HOUR()

Returns the number of hours since midnight to the present time.

MONTHDAY(Value)

Returns the day of the month associated with *Value*, where *Value* is the number of days since January 1, 1978.

MONTH(Value)

Returns the month number associated with *Value*, where *Value* is the number of days since January 1, 1978.

MINUTES()

Returns the number of minutes since midnight to the current time.

NOW()

Returns the time of day as a fraction of a day (this format is used in a time field).

TIME(Hours, Minutes, Seconds)

Returns the time values, as stored in a time field, for the given time in *Hours*, *Minutes* and *Seconds*.

TODAY()

Returns the date value, as stored in a date field, for today's date. (When entering today's date in a field, use an asterisk.)

WEEKDAY(Value)

Returns the day-of-the-week number associated with *Value*, where *Value* is the number of days since January 1, 1978. (Sunday = 0, Monday = 1, etc.)

YEAR(Value)

Returns the year number associated with *Value*, where *Value* is the number of days since January 1, 1978.

ARexx

If you own the ARexx programming language package, you can control some of **GoldFile**'s basic functions using ARexx macros. For information about the ARexx language in general, consult your ARexx documentation.

External ARexx macros can communicate with **GoldFile** through the port name "File" (specify *address 'File'* in ARexx). You can invoke ARexx macros from the CLI using the *rx* command as usual, or you can run a macro from **GoldFile** using the *Special/ARexx* menu item:

- Select *Special/ARexx*; a special file requester will appear.

- Select the ARexx macro you wish to execute from the file requester.
- Enter any parameters for the ARexx macro into the appropriate text box in the requester.
- Click *Continue* on the requester to execute the ARexx macro.

Below is a description of the ARexx commands that can be used when communicating with **GoldFile**.

- **SaveTemplate:** Saves the template to disk.
- **BrowseOn:** Turns *Browse* mode on.
- **BrowseOff:** Turns *Browse* mode off.
- **AddOn:** Turns *Add* mode on.
- **AddOff:** Turns *Add* mode off.
- **DeleteRecords:** Deletes all the highlighted records.
- **NextRecord:** Goes to the next record.
- **PrevRecord:** Goes to previous record.
- **GoTo:** Goes to specified record.

Format: GoTo
 FieldName
 RecordNumber

Example: GoTo
 "Address"
 "5"

- **UseIndex:** Use an Index.

Format: UseIndex
 Name of Index

Example: UseIndex
 "First Field"

- **CreateFilter:** Creates a new filter

Format: CreateFilter
 Name of Filter
 Formula

Example: CreateFilter
 "RexxFilter"
 'Field1="Amy*"'

- **ApplyFilter:** Apply an already existing filter

Format: ApplyFilter
 FilterName

Example: ApplyFilter
 "RexxFilter"

- **ForgetFilter:** Forget the filters.

- **DeleteFilter:** Deletes the specified filter.

Format: DeleteFilter
 FilterName

Example: DeleteFilter
 "RexxFilter"

- **LoadFilter:** Loads a filter from disk

Format: LoadFilter
 FileName

Example: LoadFilter
 "df0:RexxFilter.fil"

- **UseScreen:** Choose screen format to use

Format: UseScreen
 ScreenName

Example: UseScreen
 "Default"

- **FindRec:** Finds the next record matching the specified criteria.
 Returns 0 if no records found, or the record number.

Format: FindRec
 Formula

Example: FindRec
 "Field1="Amy*"

- **FindNext:** Finds the next record matching the formula specified
 in FindRec. Returns 0 if no records found, or the record number.

- **EditSub:** Same as *Command/Edit/Substitute*.

Format: EditSub
Field
FirstRec
LastRec
Substitute Entry

Example: EditSub
"Names"
"1"
"10"
"Peter Paul"

- **EditForm:** Same as *Command/Edit/Formula*.

Format: EditForm
Field
FirstRec
LastRec
Formula

Example: EditForm
"Balance"
"1"
"10"
"Balance - 10"

- **Purge:** Purges database of all deleted files.

- **Rebuild:** Rebuilds database

- **Current:** Returns current selected field/record, or range of records. Puts result in result.

Example: Current
may return: "Address/5"
or: "5:9"

- **DrawMessage:** Draws a message to the message box

Format: DrawMessage
Message

- **GetContents:** Gets the contents of the currently selected item.

- **PutContents:** Puts value in currently selected item.

Format: PutContents
NewContents

Example: PutContents
"685-2363"

- **GetString:** Gets a string from File's string box

Format: GetString
Message (to go in message box)
Initial String

Example: GetString
"Enter amount of Subscription:"
"22.95"

- **MaxRec:** Returns how many records visible in current filter.

- **Quick Calc functions:**

CalcSum, CalcAvg, CalcCount, CalcMax, CalcMin, CalcVar, CalcStd

All quick calc functions work in the following way:

Format: CalcSum
FieldName
FirstRow
LastRow

The result of the calculation is placed into the Arexx variable *result*.

